Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-61. (Cancelled)

62. (currently amended):

A processor comprising:

an instruction delivery engine to store and fetch instructions ~~either~~ from one or more threads based upon a current processing mode; and

an allocator to receive instructions from the instruction delivery engine and to perform allocation in a resource based upon the current processing mode.

63. (currently amended):

The processor of claim 62 wherein the allocator assigns the entire resource to the thread that is active if the current processing mode is single threading, and the allocator assigns a portion of the resource to each of the threads running concurrently if the current processing mode is multithreading.

64. (cancelled)

65. (currently amended):

The processor of claim 63 ~~wherein~~ wherein:

if the current processing mode is single threading, the allocator allocates an amount of entries for the instructions from the active thread in the resource if the resource has sufficient available entries and wherein the allocator activates at least one stall signal if the resource does not have sufficient available ~~entries.~~ entries; and,

if the current processing mode is multithreading, the allocator allocates an amount of entries for the instructions from each respective thread in the respective portion if the respective portion has sufficient available entries and wherein the allocator activates at least one stall signal if the respective portion does not have sufficient available entries.

66. (cancelled)

67. (currently amended):

The processor of claim ~~66~~ 65 wherein the instruction delivery engine uses the at least one stall signal to perform its corresponding function.

68. (original):

The processor of claim 67 wherein the instruction delivery engine re-fetches the stalled instructions in the respective thread to the allocator if the at least one stall signal is activated.

69. (original):

The processor of claim 67 wherein the instruction delivery engine fetches a subsequent instruction from another thread to the allocator if the at least one stall signal for the respective thread is activated and said another thread is not stalled.

70. (original):

The processor of claim 67 wherein the instruction delivery engine fetches an invalid instruction to the allocator if the stall signal for the respective thread is activated.

71-83. (Cancelled)

84. (new):

A method for allocating resources by a processor, the method comprising:

fetching instructions from one or more threads based upon a current processing mode; and,

performing allocation in a resource for the instructions based upon the current processing mode.

85. (new):

The method of claim 84 further comprising:

assigning the entire resource to the thread that is active if the current processing mode is single threading, and,

assigning a portion of the resource to each of the threads running concurrently if the current processing mode is multithreading.

86. (new):

The method of claim 85 wherein:

if the current processing mode is single threading,

allocating an amount of entries for the instructions from the active thread in the resource if the resource has sufficient available entries, and

activating at least one stall signal if the resource does not have sufficient available entries; and,

if the current processing mode is multithreading,

allocating an amount of entries for the instructions from each respective thread in the respective portion if the respective portion has sufficient available entries, and

activating at least one stall signal if the respective portion does not have sufficient available entries.

87. (new):

The method of claim 86 further comprising using the at least one stall signal to perform corresponding functions by the instruction delivery engine.

88. (new):

The method of claim 87 further comprising re-fetching the stalled instructions in the respective thread to the allocator if the at least one stall signal is activated.

89. (new):

The method of claim 87 further comprising fetching a subsequent instruction from another thread to the allocator if the at least one stall signal for the respective thread is activated and said another thread is not stalled.

90. (new):

The method of claim 87 further comprising fetching an invalid instruction to the allocator if the stall signal for the respective thread is activated.

91. (new):

A processor comprising:

means for fetching instructions from one or more threads based upon a current processing mode; and,

means for performing allocation in a resource for the instructions based upon the current processing mode.

92. (new):

The processor of claim 91 further comprising:

means for assigning the entire resource to the thread that is active if the current processing mode is single threading, and,

means for assigning a portion of the resource to each of the threads running concurrently if the current processing mode is multithreading.

93. (new):

The processor of claim 92 wherein:

means that are operative if the current processing mode is single threading for allocating an amount of entries for the instructions from the active thread in the resource if the resource has sufficient available entries, and activating at least one stall signal if the resource does not have sufficient available entries; and,

means that are operative if the current processing mode is multithreading for allocating an amount of entries for the instructions from each respective thread in the respective portion if the respective portion has sufficient available entries, and activating at least one stall signal if the respective portion does not have sufficient available entries.

94. (new):

The processor of claim 93 further comprising means for using the at least one stall signal to perform corresponding functions by the instruction delivery engine.

95. (new):

The processor of claim 94 further comprising means for re-fetching the stalled instructions in the respective thread to the allocator if the at least one stall signal is activated.

96. (new):

The processor of claim 94 further comprising means for fetching a subsequent instruction from another thread to the allocator if the at least one stall signal for the respective thread is activated and said another thread is not stalled.

97. (new):

The processor of claim 94 further comprising means for fetching an invalid instruction to the allocator if the stall signal for the respective thread is activated.